

**Amendments to the Specification:**

Please replace paragraph [0001] with the following rewritten paragraph [0001]:

[0001] U.S. Patent Application Serial No. 10/672,698, filed September 25, 2003 in the name of inventor Eduard K. de Jong, entitled "Application Program Obfuscation", ~~Attorney Docket No. SUN-P7008~~, commonly assigned herewith.

Please replace paragraph [0002] with the following rewritten paragraph [0002]:

[0002] U.S. Patent Application Serial No. 10/672,700, filed September 25, 2003 in the name of inventor Eduard K. de Jong, entitled "Permutation of Opcode Values for Application Program Obfuscation", ~~Attorney Docket No. SUN-040023~~, commonly assigned herewith.

Please replace paragraph [0003] with the following rewritten paragraph [0003]:

[0003] U.S. Patent Application Serial No. 10/672,183, filed September 25, 2003 in the name of inventor Eduard K. de Jong, entitled "Multiple Instruction Dispatch Tables for Application Program Obfuscation", ~~Attorney Docket No. SUN-040024~~, commonly assigned herewith.

Please replace paragraph [0004] with the following rewritten paragraph [0004]:

[0004] U.S. Patent Application Serial No. 10/673,021, filed September 25, 2003 in the name of inventor Eduard K. de Jong, entitled "Interleaved Data and Instruction Streams for

Application Program Obfuscation", ~~Attorney Docket No. SUN-040026,~~ commonly assigned herewith.

Please replace paragraph [0005] with the following rewritten paragraph [0005]:

[0005] U.S. Patent Application Serial No. 10/672,184, filed September 25, 2003 in the name of inventor Eduard K. de Jong, entitled "Rendering and Encryption Engine for Application Program Obfuscation", ~~Attorney Docket No. SUN-040027,~~ commonly assigned herewith.

Please replace paragraph [0085] with the following rewritten paragraph [0085]:

[0085] Turning now to FIG. 21A, a block diagram that illustrates a linear application program execution order is presented. As shown in FIG. 21A, the location of the next instruction to execute can be determined based at least in part on the current instruction. If the current instruction is a jump instruction (2155, 2160), the next instruction is the address specified in the jump instruction (2165, 2170). If the current instruction is not a jump instruction, the next instruction to execute is determined by advancing~~advancing~~ the current instruction address.